B2

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04Q 11/04, H04L 12/56 — A1

(11) International Publication Number: WO 99/03298

(43) International Publication Date: 21 January 1999 (21.01.99)

(21) International Application Number: PCT/SE98/01289

(22) International Filing Date: 30 June 1998 (30.06.98)

(30) Priority Data:
08/893,479 11 July 1997 (11.07.97) US

(71) Applicant: TELEFONAKTIEBOLAGET LM ERICSSON (publ) [SE/SE]; S-126 25 Stockholm (SE).

(72) Inventor: WICKLUND, Göran; Diligensvägen 82, S-131 48 Nacka (SE).

(74) Agent: ERICSSON TELECOM AB; IPR Management & Patent Dept., S-126 25 Stockholm (SE).

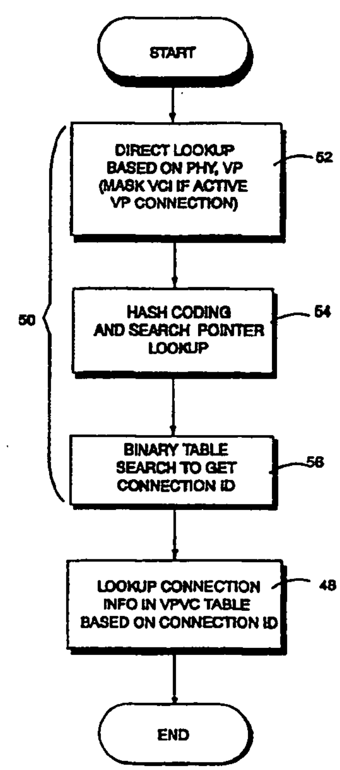
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, GW, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: VP/VC LOOKUP FUNCTION

(57) Abstract

A new, efficient approach to ATM connection table lookup minimizes the number of tables and memory lookups through use of hash coding and binary table search techniques. The virtual connection information associated with an incoming ATM cell is hash coded. The hash code provides a compressed representation of the virtual connection information, allowing the address space of a table accessed based on the hash code to be much smaller than the maximum number of possible virtual connection combinations that can be encoded in the ATM cell header without restricting the set of possible virtual connection combinations. A binary search based on the cell's virtual connection information can be used to efficiently select, from plural records accessed based on the hash code, the particular search record corresponding to the cell's connection.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

VP/VC LOOKUP FUNCTION

CROSS REFERENCE TO RELATED APPLICATIONS

This application is related to the following: U.S. Patent Application Serial Number 08/___,___ (attorney docket: 1410-321), entitled "VC MERGING FOR ATM SWITCH," filed July 11, 1997; U.S. Patent Application Serial Number 08/___,___ (attorney docket: 1410-322), entitled "ABR SERVER," filed July 11, 1997; U.S. Patent Application Serial Number 08/___,___ (attorney docket: 1410-323), entitled "HANDLING ATM MULTICAST CELLS," filed July 11, 1997; and U.S. Patent Application Serial Number 08/___,___ (attorney docket: 1410-324), entitled "A DATA SHAPER FOR ATM TRAFFIC," filed July 11, 1997.

Field Of The Invention

This invention relates to transporting data through a communications network, and more particularly, to efficient connection lookup techniques that can be used in Asynchronous Transfer Mode ("ATM") networks and switches. Still more specifically, the invention relates to a method for efficiently looking up connection information in a ATM switch Virtual Path/Virtual Channel connection table based on hashing and binary search.

Background Of The Invention

Asynchronous Transfer Mode (ATM) has emerged as the primary networking technology for next-generation digital communications. ATM provides a connection-based, fast packet switching technology that can be used to deliver many different types of information including audio, video, and data in real time.

ATM has a connection oriented protocol that associates each ATM data packet (called a "cell" in ATM terminology) with a given "virtual channel" supported by a physical link. Each connection is identified by two sub-fields: the Virtual Channel Identifier ("VCI") and the Virtual Path Identifier ("VPI"). Together, these fields are used in multiplexing, demultiplexing and switching a cell through the network. The VCI and VPI are not addresses; they are explicitly assigned at each segment (link between ATM switches) of a connection when the connection is established, and they remain for the duration of the connection.

When an ATM switch receives an incoming cell on an input port, it must determine which output port to route the cell to based on the incoming cell's VPI, VCI and input port physical identifier. The ATM switch must also determine new VPI and VCI values to substitute into the cell header so the cell will be correctly routed by the next ATM network segment. The ATM switch typically looks up this connection information in a VP/VC connection table based on the cell header's VPI and VCI information plus additional "physical layer" (PHY) information identifying the

input port. To maximize throughput of cells through the ATM switch, it is important to provide a rapid and efficient technique and arrangement for accessing the proper connection information from the connection table.

Under international ATM standards, the virtual path identifier (VPI) and virtual channel identifier (VCI) fields have the following sizes:

- the VPI field is twelve bits long; and
- the VCI field is sixteen bits long.

A typical size for the physical link identifier (PHY) is 5 bits. Thus, the total length of these three fields together is on the order of thirty-three bits – corresponding to about 8.9 billion (2^{33}) combinations. However, the maximum number of connections a particular ATM switch can set up is typically restricted to a much lower value (for example, 16K). In the most general case, it should be possible to use any arbitrary sub-set of the total number of 8.9 billion PHY, VPI, VCI combinations as identifiers for the channels that the ATM switch sets up. The problem is to find an efficient mechanism that can take the PHY, VPI, VCI field (e.g., 33 bits long), find out whether it is a valid combination of bits, and output a pointer to one of a smaller (e.g., 16K) connections in a virtual path/virtual channel connection table sized to accommodate the number of connections the ATM switch can maintain.

U.S. Patent No. 5,414,701 to Shtayer et al provides one approach to solving this problem. The Shtayer et al. patent

describes an address compression method that relies on a hierarchical arrangement of separate link, virtual path, and virtual channel tables. In Shtayer et al, the PHY information indexes into a "link" table that provides a virtual path pointer. The virtual path pointer is used in combination with the VPI from the cell header to index into one of many virtual path tables. The virtual path table provides a virtual channel offset that is, in turn, used in combination with the cell header's VCI to index into one of many virtual channel tables. The virtual channel tables provide ingress connection identifiers that are used to physically route data through an ATM system.

While Shtayer et al's approach may avoid the need for a large, mostly unused connection table, its process of using link information to index a link table, using information looked up in the link table to index a virtual path table, etc. is unduly time consuming and inefficient. Furthermore, it may be difficult to maintain Shtayer et al's complex hierarchically structured, separate link, virtual path and virtual channel tables.

U.S. Patent No. 5,481,687 to Goubert et al. uses masking to form subsets of reduced addresses that may then be translated to provide a non-overlapping series of relative addresses. Goubert's technique requires a complex use of different masks for selecting different address bits, and may suffer from flexibility problems.

Summary of the Invention

The present invention provides a new, efficient approach to ATM connection table lookup that minimizes the number of tables and table lookups through use of hash coding and binary table search techniques.

In accordance with one aspect provided in accordance with the present invention, the PHY, VPI and VCI virtual connection information associated with an incoming ATM cell are hash coded to produce a hash code of reduced length. The hash code is used to index into a table sized to correspond to at least the maximum number of active connections the ATM switch is capable of sustaining. The hash code provides a compressed representation of the PHY, VPI, VCI information – allowing the table address space to be much smaller than the maximum number of PHY, VPI, VCI combinations without restricting the set of PHY, VPI, VCI combinations that can be used.

Because the hash coding results in a loss of information, there is no one-to-one correspondence between PHY, VPI, VCI combinations and hash codes. Rather, the hash coding will transform plural PHY, VPI, VCI values into the same hash code. In accordance with another aspect provided by this invention, a search table can maintain a number of different search records corresponding to each hash code. In the preferred example, the search table selects a particular record to retrieve based on the

hash code and the PHY, VPI, VCI information associated with the cell.

A binary search is used in the preferred example to efficiently locate the particular search record in the search table. In the preferred example, the search table maintains search records in a binary tree structure. A binary search technique efficiently locates the particular search record corresponding to the ATM cell's VPI/VCI and the PHY physical link identifier.

The present invention thus provides an efficient virtual path/virtual channel table lookup based on hashing and binary search that does not restrict the set of PHY, VPI, VCI combinations used, and minimizes the number of memory accesses.

Brief Description of the Drawings

These and other features and advantages provided by the present invention will be better and more completely understood by referring to the following detailed description of presently preferred example embodiments in connection with the drawings, of which:

Figure 1 shows an example of how VPI, VCI and PHY virtual connection information is used in an overall Virtual Path/Virtual Channel lookup process;

Figure 2 shows an example simplified process for efficiently looking up VP/VC information based on hash coding and binary search;

Figure 3 shows an example VP/VC binary search tree;

Figure 4 shows an example VP/VC lookup arrangement;

Figure 5 shows an example connection setup process; and

Figure 6 shows an example connection removal process.

Detailed Description Of Presently Preferred Example Embodiments

Figure 1 shows an example overall VP/VC lookup process that may, for example, be performed by an ATM switch. In this example, an ATM cell 10 incoming from a segment of ATM network 20 includes a standard header 12 and a data payload 14. Header 12 includes virtual connection information, i.e., a 12-bit virtual path identifier (VPI) 16 and a 16-bit virtual channel identifier (VCI) 18. In addition, a 5-bit PHY field 22 in this example indicates physical interface information (e.g., the ATM switch input port identifier) associated with ATM cell 10.

In this example, VPI 16, VCI 18 and PHY 22 are used to access connection information stored in a virtual path/virtual channel connection or routing table 100. The ATM switch uses connection information from routing table 100 to route ATM cell 10 to a switch output port, and also may obtain new VPI and VCI fields from the routing table to substitute into cell header 10 for use by the next segment of ATM network 20.

In this example, the maximum number of connections the ATM switch can sustain at any given time is 16K. Therefore,

connection table 100 stores a maximum of 16K (2^{14}) entries -- one for each different connection the ATM switch can maintain. Different ATM switches can maintain different numbers of connections; in general, connection table 100 is sized to match the capabilities of the particular ATM switch it is used with. In this particular example, since connection table 100 stores 16K entries, a 14-bit long channel identifier is required to uniquely identify a particular connection record within the connection table.

Simplified Look Up Process

In this example, a lookup function 50 provides, based on the combination of PHY 22, VPI 16 and VCI 18, a 14-bit long channel identifier for looking up entries within connection table 100. Lookup function 50 in this example consists of three discrete steps shown in the Figure 2 example simplified lookup process:

1. PHY, VP direct lookup (step 52),
2. Hash-coding and search pointer lookup (step 54), and
3. Table search (step 56).

The result of the table search (step 56) is a 14-bit connection or channel ID that is used to look up connection information in the VP/VC connection table 100 (step 58).

On a simplified level, the direct lookup step 52 determines whether the PHY 22 and VPI 16 combination corresponds to an active virtual path connection (VPC). If it does, then the VCI field 18 in the incoming ATM cell 10 is irrelevant to the lookup – i.e., it could have any value that should be switched through without

modification. Accordingly, if the direct lookup step 52 detects that the PHY 22, VPI 16 information corresponds to an active virtual path connection, the direct lookup step masks the VCI field 18 prior to performing the following hash coding and search pointer lookup step 54.

Briefly, the purpose of the hash coding and search pointer lookup step 54 is to generate a reduced-length hash code from the PHY, VPI, (masked) VCI combination. In the preferred example, the resulting hash code is used, directly or indirectly, by search step 56 to look up a channel identifier stored in a search table. In this preferred example, the hash code produced by step 54 is used as an address into a second look-up table that, for each position, contains a pointer into the search table. Step 56 uses this search pointer (plus the cell's virtual connection information) to look up corresponding connection information in the search table.

Since the hash coding performed by step 54 results in a loss of information, it may happen that many active connections will be hash coded into the same value. In that case, search step 56 must perform a search among all connections with the same hash code. In the preferred example, step 56 uses a binary search to minimize the number of accesses into the search table. Figure 3 shows an example VP/VC binary search tree that may be maintained in the search table and used for the binary search performed by step 56.

More Detailed Example VP/VC Lookup System

Figure 4 shows an example virtual path/virtual channel lookup system 200 that performs the lookup process 50 shown in Figure 2. In this example:

- a VPC lookup/VCI mask block 202 performs the direct lookup step 52 of Figure 2,
- a hash coding block 204 and a pointer lookup block 206 perform Figure 2 step 54, and
- a search block 208 performs Figure 2 search step 56.

In this example, both VPI 16 and PHY 22 are inputted to the VPC lookup/VCI mask block 202. Block 202 takes the PHY, VPI combination (seventeen bits) and reads out a 1-bit data information indicating whether that combination is an active virtual path connection ("VPC") or not. This single table 202a does not provide link information, but rather, outputs active virtual path connection information based on both VPI and PHY. In more detail, block 202 of this example includes a direct lookup memory 202a comprising a 128K-by-1 bit RAM. Direct lookup memory 202a may, for example, provide a single bit output for each possible PHY, VPI combination indicating if the combination is an active virtual path connection ("VPC") or not ("NO").

As is well known, there are two types of connections that can be active: active Virtual Path Connection ("VPC") and active Virtual Channel Connection ("VCC"). A VPC is identified by the VPI-field (and the PHY); the VCI field is irrelevant for the switch

in this case. All cells belonging to an active VPC are switched the same way, so in order to find the right entry in the channel table, the VCI field must be masked. This is done after a check in the PHY, VPI table 202a in this example.

A VCC is identified by the PHY, the VPI and the VCI. In this case, cells 10 with different VCIs (but the same PHY and VPI) belong to different connections. The switch may simultaneously support VPCs and VCCs.

As described above, the first table 202a (PHY, VPI) in this example tells whether the cell belongs to an active VPC -- in which case the VCI field is masked. If it is not an active VPC, it could either be an active VCC or an illegal VPI, VCI combination. Search table 208a is used to make this latter determination in this example.

Thus, in this example, if the PHY, VPI information of a cell 10 corresponds to an active virtual path connection, the VCI field 18 in the cell is irrelevant (i.e., it could have any value that should be switched through without modification). In the case of an active virtual path connection, a multiplexer 202b (or other hardware or software masking function) masks the VPI field 16 to all 0's before using the bit field in the next stage, i.e., PHY, VPI, 0. The VCI field 18 is not masked in this example if the VP connection is not active. The resulting, selectively masked VCI field 18 outputted by multiplexer 202b is designated as VCI' on Figure 4.

In this example, the masking function described above will also mask all VCI ≤ 31 (or 15) to 0. The reason for this is that these VCI-values are reserved for carrying Operation and Maintenance information related to the Virtual Path i.e., these VCI values can not be used for Virtual Channels. When such a cell 10 arrives, it is treated as belonging to a Virtual Path -- even if the VPC/VCI look-up table 202a says that this Virtual Path is not active.

Hash coding block 204 in this example generates a 14-bit hash code from the 33-bit PHY, VPI, (masked) VCI′ combination. In this particular example, hash coding block 204 generates a 14-bit long hash code that represents, in a reduced (compressed) length format, the PHY, VPI, VCI′ combination. One way of doing this is to generate a CRC 14 value from the 33-bit value. Formally, this is done by appending fourteen zeros at the end of the 33-bit value, i.e., generating a 47-bit value, treating this as a polynomial of degree forty-seven, and dividing this by a generating polynomial of degree fourteen. An example of a generating polynomial is $x^{14} + x^{13} + x^{11} + x^{9} + 1$. Many others will work as well. The CRC is the remainder left when the division has been completed. One efficient way to perform this particular CRC 14 hash coding is to use a 14-bit feedback register 204a and a complex network of many XOR gates 204b. Those skilled in the art will understand that other hash coding arrangements and/or implementations could be used.

The hash code generated by hash coding block 204 is, in this example, used to address into a second lookup table 206 that for each position contains a 14-bit pointer into search table 208. In this example, pointer lookup table 206 has 16K entries each fourteen bits long – for a total size of 224K bits. If desired, the table 206 could be larger than the maximum number of connections, e.g., 32k (for 16k connections). The advantage with this is that the probability for a full binary tree will be reduced, as there are more hash codes available (more trees).

Search block 208 includes a search table 208a that stores search records 209 corresponding to active ATM switch connections. Because the hash coding process performed by hash coding block 204 results in the loss of information, it may happen that many active connections will be hash-coded into the same pointer value generated by pointer lookup 206. In that case, search block 208 must search among all connections within memory 208a with the same hash code to find the one search record 209 corresponding to the cell 10's connection. In this example, search table 208 uses a binary search to minimize the number of steps in the search (e.g., number of accesses into search table 208a).

In theory, all 16K connections could have been hash-coded to the same value, which means that the binary search must be fourteen levels deep. However, since the probability of this is negligible, a more abbreviated search can be used in practical ATM switch implementations. For example, a 4-level search will

allow fifteen PHY, VPI, VCI′ combinations to be hashed to the same code. The consequence of this is that during the process of establishing a connection, a sixteenth (free) PHY, VPI, VCI′ value hashed to a value that fifteen other active connections have already been hashed to, cannot be used. The probability of this happening is very small: approximately $\frac{N}{e(n+1)!}$, where n is the size if the tree and N is the size of the search table (e=2.718..). With n=15 and N=16K, this is below 10^{-9} (simulations may be used to verify this). Thus, although there is some blocking probability (i.e., a free PHY, VPI, VCI combination can not be used due to a full binary search tree), this can be neglected for all "normal" usages of PHY, VPI, VCI.

In the example embodiment, each search record 209 in search table 208 includes the following fields:

- PHY/VPI/VCI combination field 210a (thirty-three bits),
- Channel identifier field 210b (fourteen bits),
- Low pointer field 210c (fourteen bits), and
- High pointer field 210d (fourteen bits).

The size of search table 208a in this example is thus 16K times seventy-five bits, i.e., 1.2 megabits.

When the VPI, VCI is used in the binary search, it is either found or not. If it is found, the VCC is active; if it is not found, the VCC is not active. An overall combined effect of cascaded tables 202a and 208a can be summarized as follows:

<u>Active VPC</u>: the PHY, VPI combination is found in the first table 202a; VCI is masked; hash and binary search gives the connection number (entry in connection table 208a)

<u>Active VCC</u>: the PHY, VPI combination is not found to be active in the first table 202a (= no VPC). The PHY, VPI, VCI combination is found after hash and binary search, and the connection number is read out from search table 208a.

<u>Illegal VPI; VCI</u>: the PHY, VPI is not found in the first table 202a (=no VPC). The PHY, VPI, VCI combination is not found after hash and binary search in the search table 208a.

Thus, in a sense, in this example a VPC is found active directly in the first table 202a but an active VCC is found active indirectly after the hash and binary search (a "hit" in the search) in the search table 208a.

Referring to Figures 3 and 4, when search block 208 performs a search, it uses the 14-bit pointer generated by pointer lookup 206 to access the "root" search record 209(1) of a binary search tree containing one to fifteen records 209(1)-209(15). A compare block 208b compares the PHY 22, VPI 16, (masked) VCI′ information associated with ATM cell 10 with the PHY/VPI/VCI′ combination field 210a within the accessed search record 209. If the PHY, VPI, (masked) VCI′ combination associated with the cell 10 is the same as the value stored in the

accessed search record 209, search logic 208c stops the search and the channel identifier 210b from the record is read out. If compare block 208b determines that the PHY, VPI, VCI′ value associated with the cell 10 is lower than the one stored in the accessed search record 209, the low pointer 210c is read out and used to access another (particular sorted) search record 209(2) of the binary search tree. If compare block 208b determines that the PHY, VPI, VCI′ value associated with cell 10 is higher than the value of field 210a of the accessed search record, search logic 208c reads out the high pointer 210d and uses it to access another (particular sorted) search record 209(9) of the binary search tree. If four searches have been made without any hit, search logic 208c generates an "illegal cell" (error) signal, and cell 10 is discarded.

Once the appropriate record 209 containing a matching PHY, VPI, VCI′ value has been found, the channel identifier 210b of that record is used as an entry into the VP/VC connection table 100 where all information relevant to the connection is stored.

Two PHY, VCI, VPI combinations hashed by hash coding block 204 into the same hash code will provide different 19-bit quotients of the CRC polynomial division described above. To reduce the size of search table 208a, it is possible to store the quotient (nineteen bits) of the CRC polynomial division instead of the complete 33-bit PHY/VCI/VPI combination within the search records 209 to save memory space. This alternative approach will require some additional (large) number of XOR gates in hash coding block 204 to output the quotient.

Example Software Driver Functions

The correct information must be stored in the various tables 202a, 206, 208a when a new connection is set up. Figure 5 shows an example connection software driver connection setup. The steps shown in Figure 5 may be performed in any desired order to accomplish the same result.

As shown in Figure 5, the driver (or some other ATM switch call routing process) must store the appropriate connection information into VP/VC connection table 100 (Figure 5, block 302). The driver software must calculate the same 14-bit hash code that hash coding block 204 will calculate based on the PHY, VPI, VCI' combination (the driver may perform this calculation in software and/or by using hardware support) (Figure 5, block 304). The driver may then determine whether a channel with the same hash code already exists within search table 208a (Figure 5, decision block 306). If a channel with the same hash code already exists ("yes" exit to decision block 306), the driver must add a new "leaf" to an existing binary search tree already stored in search table 208a (Figure 5, block 308). If a channel with the same hash code does not already exist within search table 208a, the driver must add a new search tree "root" to the search table (Figure 5, block 310), and store a new hash code pointer into the pointer lookup table 206 (Figure 5, block 312). The driver must also update the VPC table 202a to indicate that a new connection has been established (Figure 5, block 314).

In this example, the driver is responsible for maintaining the binary search tree in a sorted condition so that the binary search will work. Thus, to add a new search tree leaf (block 308), the driver in this example searches through search table 208a with the new PHY, VPI, VCI combination, and inserts the new leaf at the correct (sorted) position within the existing search tree. It may happen that the search tree becomes unbalanced, depending upon the history of connection set ups and tear downs. For example, the PHY, VPI, VCI combination associated with a "root" record 209 for a particular search tree may have a relatively high or relatively low value. This means that the driver will not be able to find a place for the new combination even though the binary search tree is not complete (i.e., it has less than fifteen branches and leaves). In this case, the driver should balance the tree by first sorting the entries linearly and then building up the tree by taking the middle entry as the root and continuing with the upper and lower subtrees. As the number of entries is fifteen or less, this procedure should not take very long. Once the new search tree is created, the driver can place the new tree at a different (free) location in search table 208a, together with the correct channel identifiers 210b. When this has been done, the driver updates the pointer in lookup table 206 to point to the new root. After this, the old tree can be removed or overwritten.

Figure 6 shows an example software driver connection removal process. When the driver removes a connection, it reconfigures the search tree -- but this is not so complicated. If the

removed connection is located at a leaf ("yes" exit to decision block 324), the leaf is simply removed (block 326). If the connection removed is at a branch, the branch is removed and is replaced by the lowest connection in the upper tree or the highest connection in the lower tree. For example, referring to Figure 3, removal of leaf 209(2) may involve replacing that leaf with leaf 209(5) or 209(7). The remaining subtrees do not have to be reconfigured. If the root is removed and there are more leaves, then a leaf replaces the root. If there are no more leaves and removal of the root is required ("yes" exit to decision block 324), the process deletes the search tree root record 209(1) (Figure 6 block 328), and deletes the associated pointer within pointer lookup 206 (Figure 6 block 330). The driver (or other call routing process) may also update the VPC table 202a to indicate that the connection is no longer active (Figure 6 block 332).

* * * * *

The present invention thus provides a new, efficient approach to ATM connection table lookup that minimizes the number of tables and memory lookups through use of hash coding and binary table search techniques. The virtual connection information associated with an incoming ATM cell is hash coded. The hash code provides a compressed representation of the virtual connection information – allowing the address space of a table the hash code accesses to be much smaller than the maximum number of possible virtual connection combinations that can be encoded in

the ATM cell header without restricting the set of virtual connection combinations that may be used. A binary search can be used to efficiently locate the particular search record corresponding to the cell's connection – providing relatively few memory accesses (e.g., 1 + 1 + 4*2=10).

While the invention has been described in connection with what is presently considered to be the most practical and preferred embodiment, it is to be understood that the invention is not to be limited to the disclosed embodiment, but on the contrary, is intended to cover various modifications and equivalent arrangements included within the spirit and scope of the appended claims.

WHAT IS CLAIMED IS:

1. In an ATM switch of the type that routes cells from an input port to an output port over a virtual connection based at least in part on virtual connection information provided within a cell header, a virtual path/virtual channel lookup process including the following steps:

(a) hash coding cell virtual connection information including Virtual Path, Virtual Channel and physical link information to provide a reduced length hash code; and

(b) accessing connection information based at least in part on the reduced length hash code.

2. A method as in claim 1 wherein hash coding step (a) includes the step of hash coding at least a virtual path identifier and a physical link identifier.

3. A method as in claim 1 wherein accessing step (b) includes the step of performing a binary search based at least in part on the hash code.

4. A method as in claim 1 wherein hash coding step (a) includes testing whether the virtual connection information corresponds to an active virtual path, and selectively masking at least a part of the virtual connection information based on the results of the test.

5. A method as in claim 4 wherein the selectively masking step comprises selectively masking a virtual channel identifier.

6. A method as in claim 1 further including the step of looking up a search table pointer based on the hash code, and accessing step (b) includes accessing a search table based at least in part on the search pointer.

7. A method as in claim 1 wherein the hash coding step (a) includes performing a CRC polynomial division.

8. A method as in claim 1 wherein hash coding step (a) includes processing the virtual connection information at least in part with a feedback register and a network of XOR gates.

9. A method as in claim 1 wherein accessing step (b) includes performing a search based on the hash code and the virtual connection information.

10. A method as in claim 1 wherein accessing step (b) includes searching a search table for search records indexed by the hash code that contain matching virtual connection information.

11. A method as in claim 1 step (b) includes selecting plural records corresponding to the hash code, and selecting one of the plural records based on the virtual connection information.

12. A method as in claim 1 wherein step (a) includes generating a hash coding byproduct, and step (b) includes

selecting plural records corresponding to the hash code, and selecting one of the plural records based on the hash coding byproduct.

13. A method as in claim 1 wherein accessing step (b) includes performing a multi-level binary search, and generating an error if a corresponding connection record is not found after a predetermined number of binary search levels.

14. A method as in claim 1 further including the step of balancing a binary search tree, and step (b) is performed by searching the balanced binary search tree.

15. A method as in claim 1 wherein the method further includes:

determining whether there is an active VPC based at least in part on both PHY and VPI virtual connection information; and

step (b) includes the step of determining whether there is an active VCC based on a search table indexed by the hash code.

16. In an ATM switch of the type that routes cells from an input port to an output port over a virtual connection based at least in part on virtual connection information providing within a cell header, a virtual path/virtual channel lookup arrangement comprising:

a hash coding block that hash codes cell virtual connection information to provide a reduced length hash code; and

a search table that accessing connection information based at least in part on the reduced length hash code.

17. An arrangement as in claim 16 wherein the hash coding block includes a feedback register and a gate array.

18. An arrangement as in claim 16 further including means for selectively masking a portion of the virtual connection information based on whether the virtual connection information corresponds to an active virtual path.

19. An arrangement as in claim 16 wherein the search table includes compare logic that compares the virtual connection information with at least portions of search records indexed by the hash code.

20. An arrangement as in claim 16 wherein the search table includes means for storing a binary search tree.

21. An arrangement as in claim 16 wherein the search table includes a memory that stores plural search records each including a virtual connection information field, a connection identifier field, a low field and a high field.

22. An arrangement as in claim 16 wherein the hash coding block hash codes a Virtual Path Identifier, a Virtual Channel Identifier and physical link information

23. An arrangement as in claim 16 wherein the arrangement further includes a circuit that determines whether the virtual connection information corresponds to an active virtual path connection; and the search table is used to determine whether the virtual connection information corresponds to an active Virtual Channel connection.

24. A virtual path/virtual channel lookup arrangement comprising:

hash coding means, connected to receive cell virtual connection information, for hash coding the cell virtual connection information to provide a reduced length hash code; and

connection information accessing means, coupled to receive the reduced length hash code, for accessing connection information based at least in part on the reduced length hash code.

25. An arrangement as in claim 22 wherein the hash coding means comprises means for hash coding a Virtual Path Identifier, a Virtual Channel Identifier and physical link information.

26. A cell routing method including:

(a) obtaining virtual connection corresponding to a cell;

(b) hash coding at least a part of the virtual connection information to provide a reduced length hash code;

(c) using the reduced length hash code, directly or indirectly, to look up a binary search tree stored in a search table, the binary search tree being capable of including plural search records;

(d) selecting one of the plural search records based at least in part on a comparison between the search record contents and the obtained virtual connection information;

(e) obtaining virtual connection information based on the selected search record; and

(f) routing the cell through a digital communications network based at least in part on the virtual connection information obtained by step (e).

27. A cell routing method wherein the hash coding step (a) comprises hash coding at least portions of VPI, VCI and PHY information.

28. An ATM cell routing determination method comprising:

(a) looking up a record in a first table based on a combination of both PHY, VPI virtual connection information;

(b) looking up a record in a second table based on a combination of the PHY, VPI virtual connection information and VCI virtual connection information;

(c) determining based on the first table look up whether the cell has an active virtual path connection; and

(d) determining based on the second table look up whether he cell has an active Virtual Channel connection.

29. A method as in claim 28 wherein step (b) comprises accessing the second table based on a selectively masked VCI hash coded with PHY and VPI.

30. A method as in claim 28 further including routing the cell based on information derived from the record looked up in the second table.

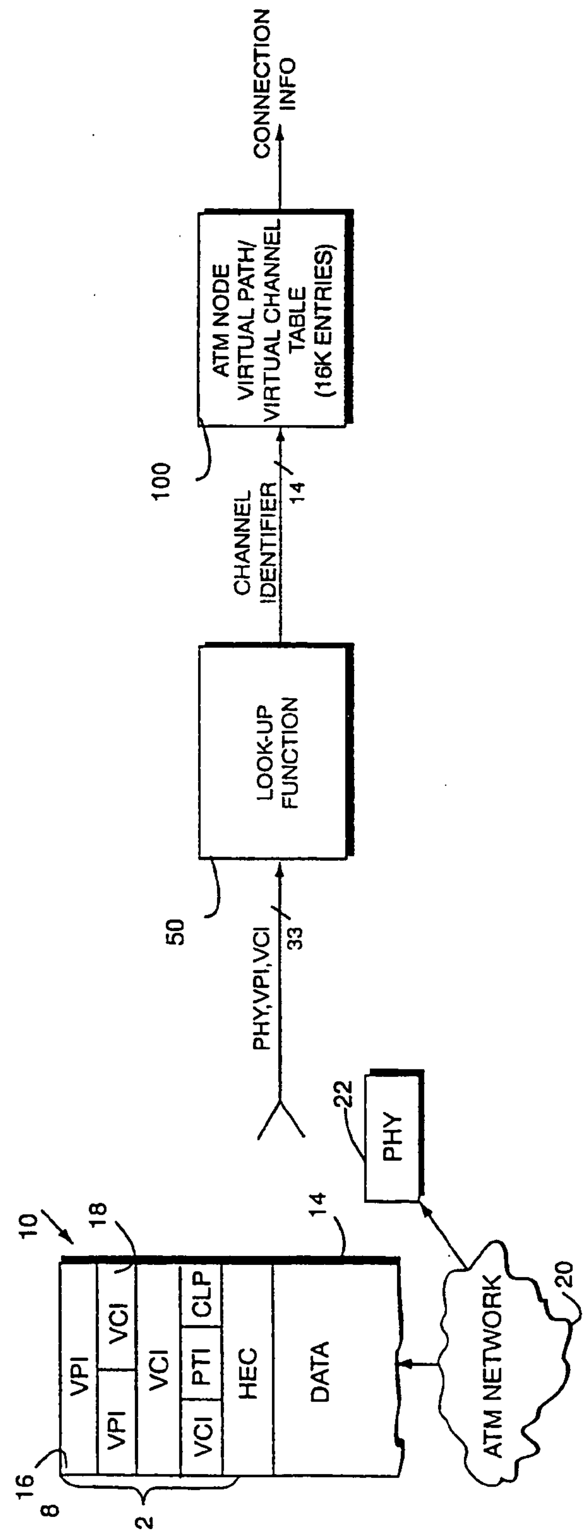

CONNECTION INFO
ATM NODE VIRTUAL PATH/ VIRTUAL CHANNEL TABLE (16K ENTRIES)
100
CHANNEL IDENTIFIER
14
LOOK-UP FUNCTION
50
PHY,VPI,VCI
33
22
PHY
10
18
14
VPI
VPI
VCI
VCI
VCI
PTI
CLP
HEC
DATA
16
8
12
ATM NETWORK
20

FIG.1

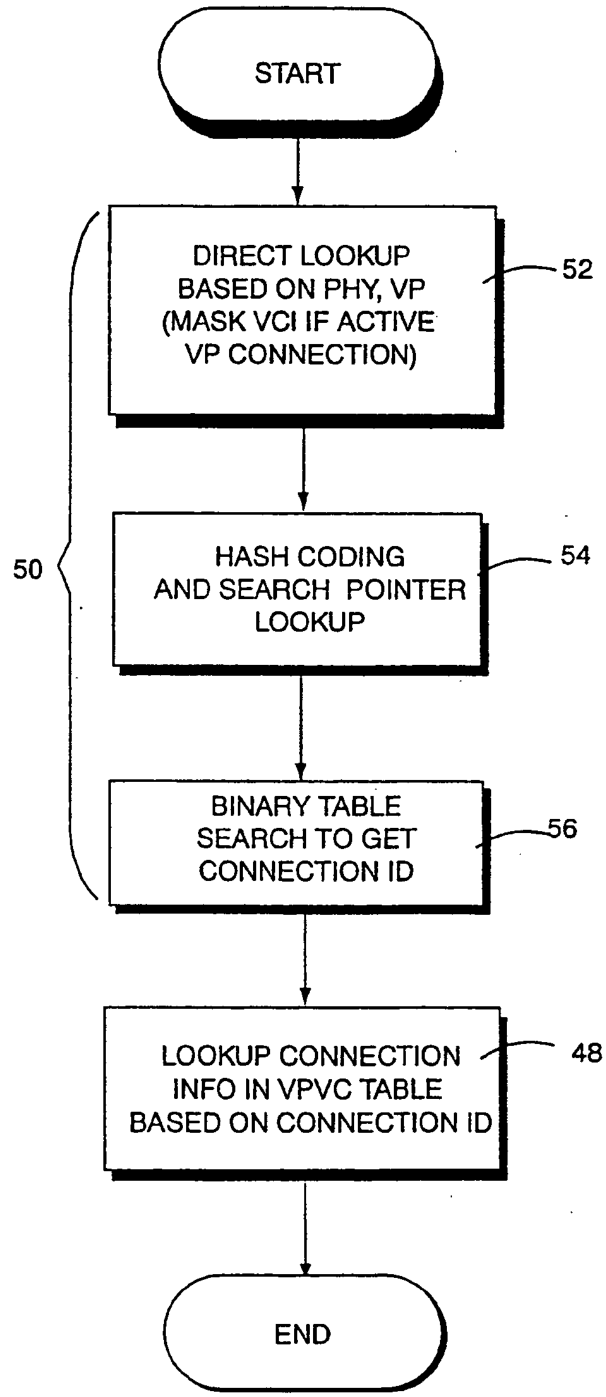

START
DIRECT LOOKUP BASED ON PHY, VP (MASK VCI IF ACTIVE VP CONNECTION)
52
50
HASH CODING AND SEARCH POINTER LOOKUP
54
BINARY TABLE SEARCH TO GET CONNECTION ID
56
LOOKUP CONNECTION INFO IN VPVC TABLE BASED ON CONNECTION ID
48
END

FIG. 2

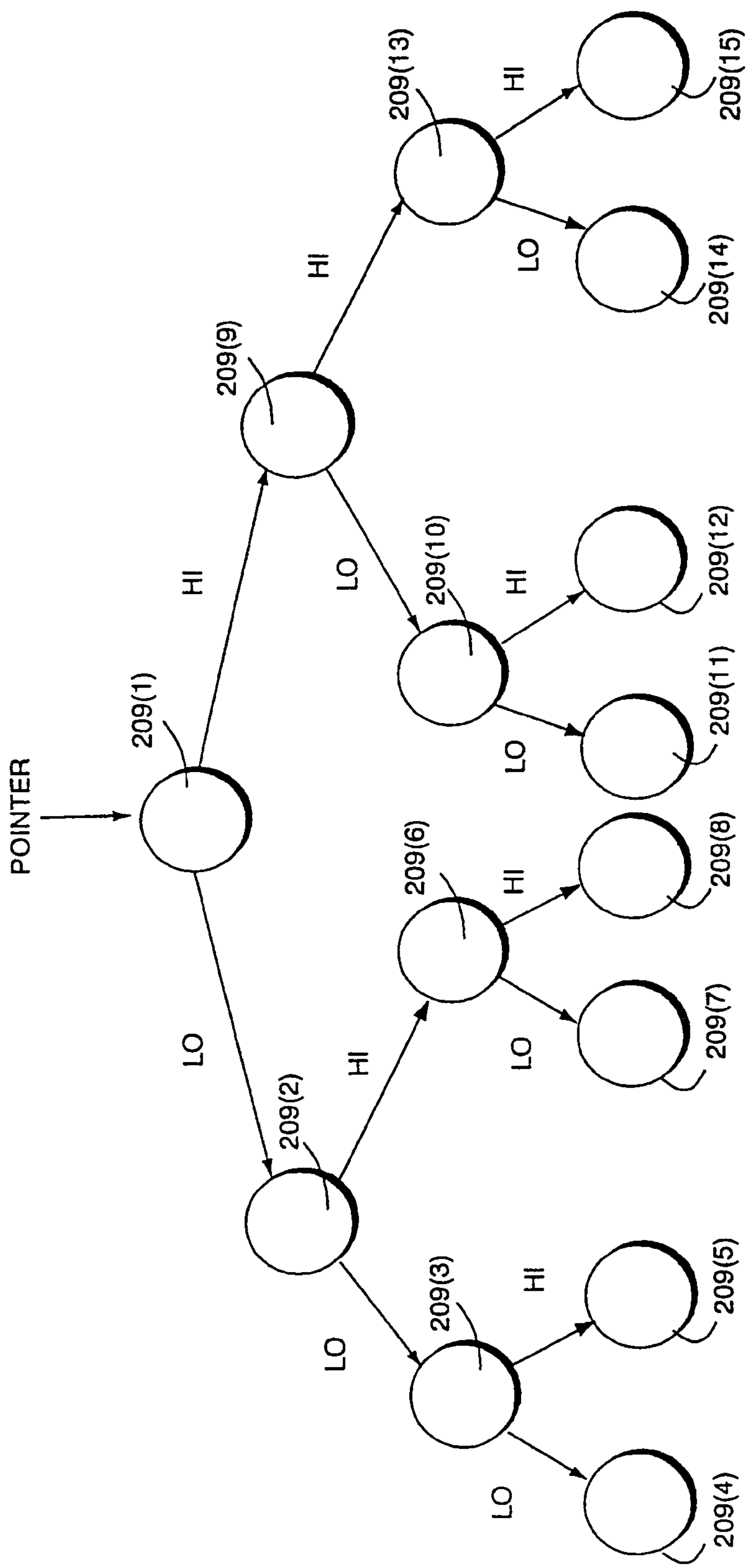

POINTER
209(1)
209(2)
209(3)
209(4)
209(5)
209(6)
209(7)
209(8)
209(9)
209(10)
209(11)
209(12)
209(13)
209(14)
209(15)
HI
LO

FIG. 3

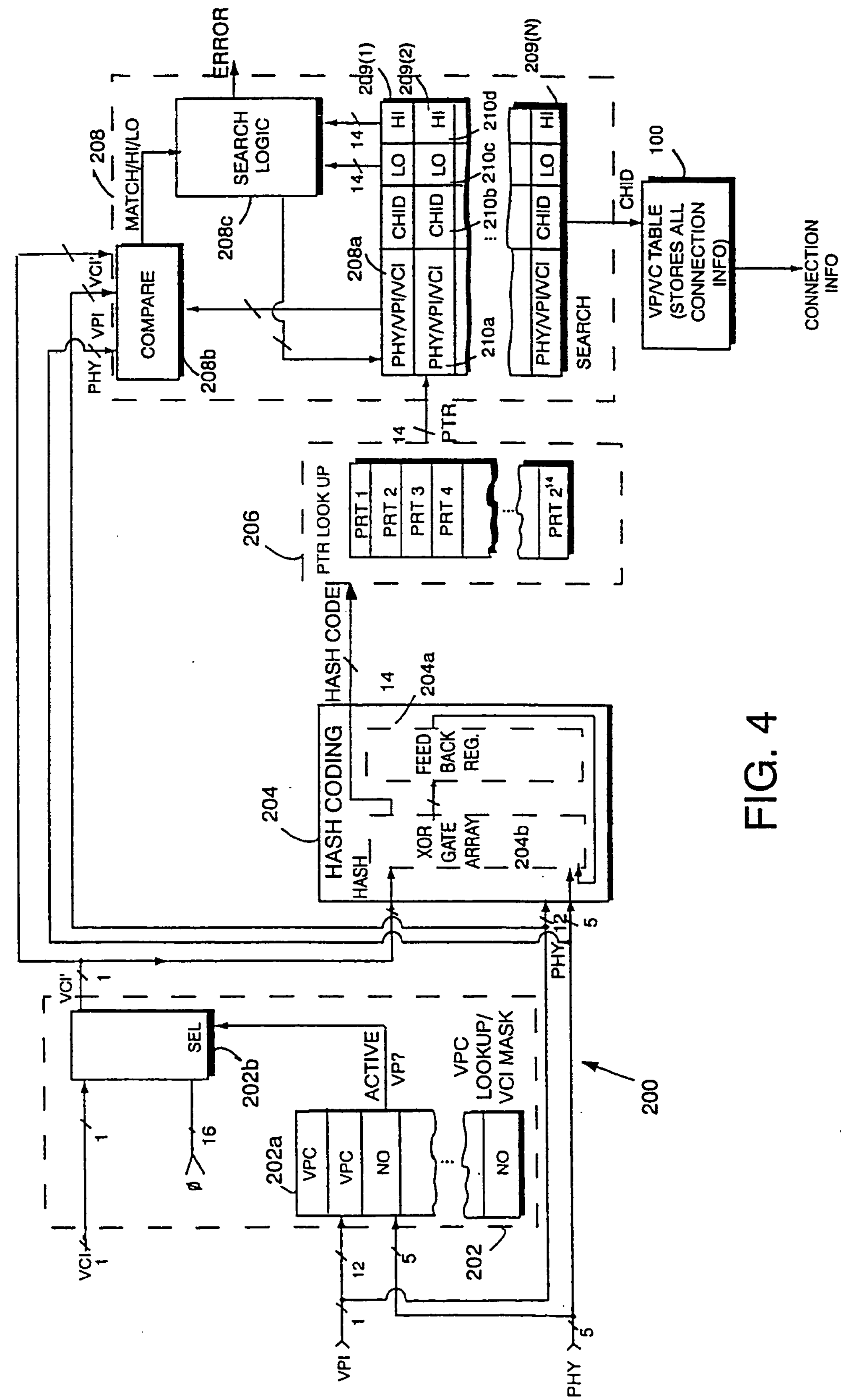

ERROR
208
MATCH/HI/LO
SEARCH LOGIC
208c
COMPARE
PHY
VPI
VCI'
208b
208a
209(1)
209(2)
209(N)
PHY/VPI/VCI
CHID
LO
HI
14
210a
210b
210c
210d
SEARCH
CHID
100
VP/VC TABLE (STORES ALL CONNECTION INFO)
CONNECTION INFO
PTR
206
PTR LOOK UP
PRT 1
PRT 2
PRT 3
PRT 4
PRT 2^14
HASH CODE
204
204a
204b
HASH CODING
HASH
XOR GATE ARRAY
FEED BACK REG.
12
5
VCI'
SEL
202b
ACTIVE VP?
VPC LOOKUP/ VCI MASK
202a
VPC
NO
202
200
16
VCI
VPI
PHY

FIG. 4

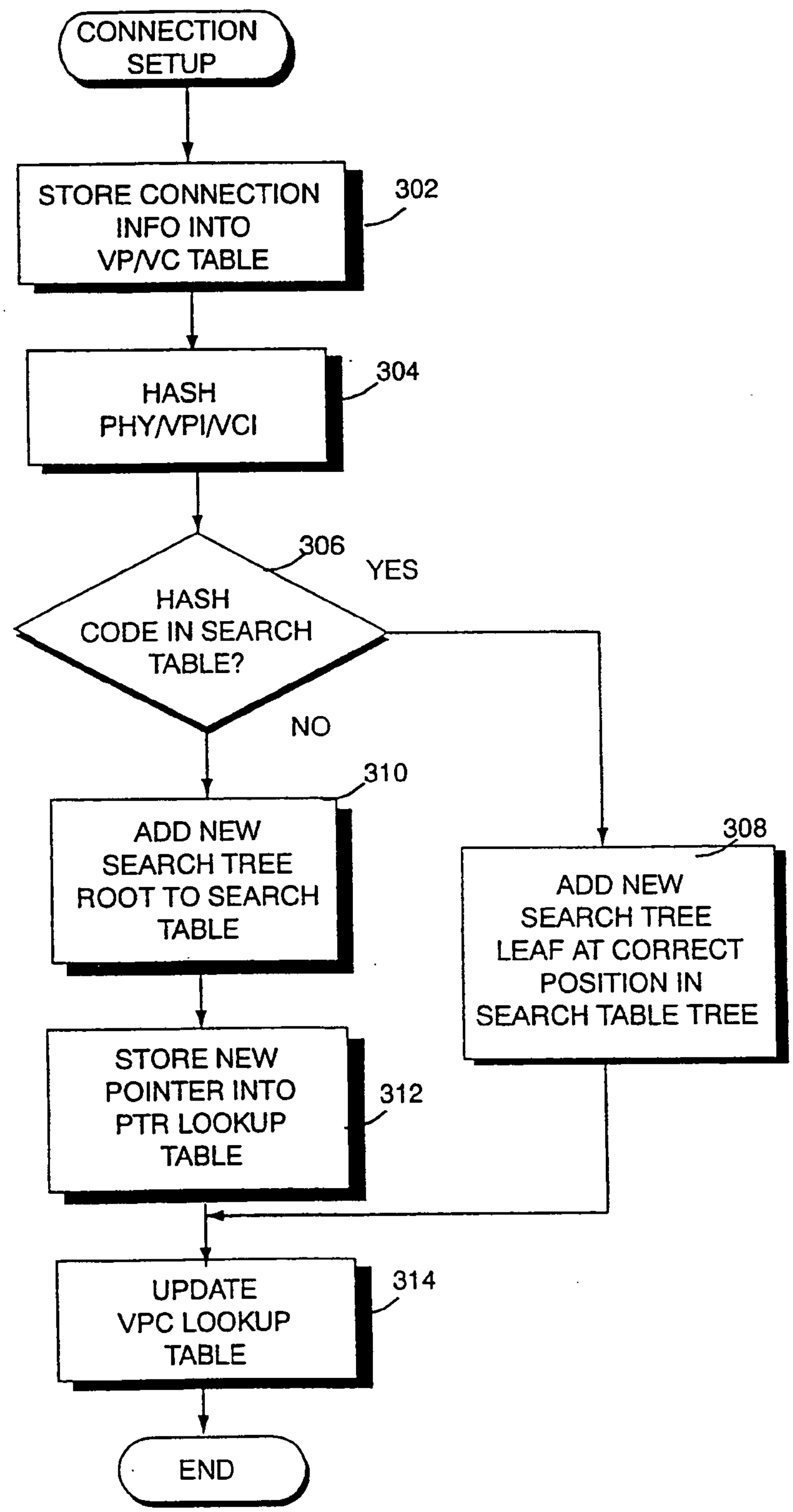

CONNECTION SETUP
STORE CONNECTION INFO INTO VP/VC TABLE
302
HASH PHY/VPI/VCI
304
306
HASH CODE IN SEARCH TABLE?
YES
NO
310
ADD NEW SEARCH TREE ROOT TO SEARCH TABLE
308
ADD NEW SEARCH TREE LEAF AT CORRECT POSITION IN SEARCH TABLE TREE
STORE NEW POINTER INTO PTR LOOKUP TABLE
312
UPDATE VPC LOOKUP TABLE
314
END

FIG. 5

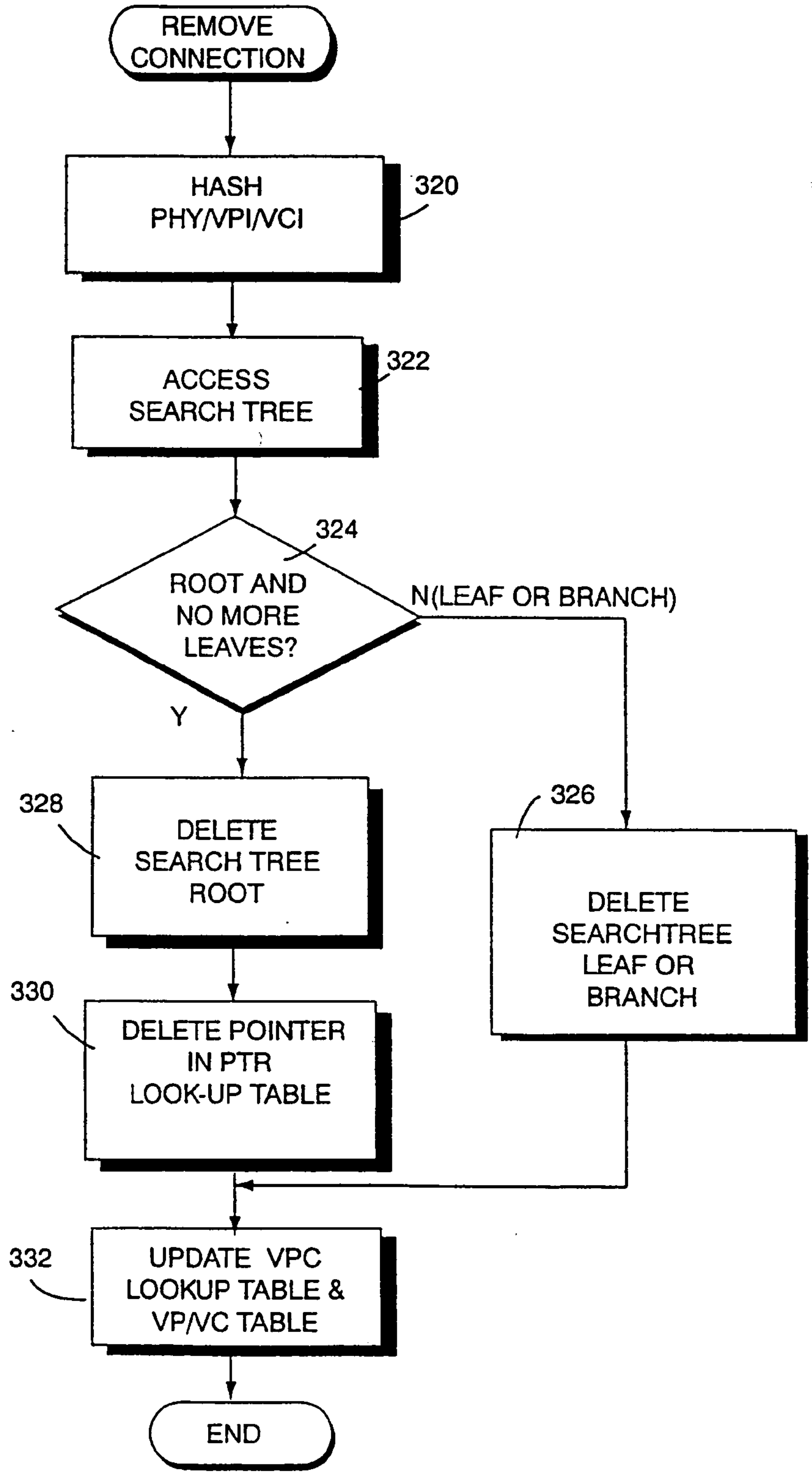

REMOVE CONNECTION
HASH PHY/VPI/VCI
320
ACCESS SEARCH TREE
322
324
ROOT AND NO MORE LEAVES?
N(LEAF OR BRANCH)
Y
328
DELETE SEARCH TREE ROOT
326
DELETE SEARCHTREE LEAF OR BRANCH
330
DELETE POINTER IN PTR LOOK-UP TABLE
332
UPDATE VPC LOOKUP TABLE & VP/VC TABLE
END

FIG. 6

INTERNATIONAL SEARCH REPORT

International Application No
PCT/SE 98/01289

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04Q11/04 H04L12/56

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04Q H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 97 13377 A (ADVANCED TELECOMMUNICATIONS MO) 10 April 1997	1,2,4,5, 15,24, 27-30
Y	see abstract	6-11, 16-19, 22,23, 25,26
	see page 6, line 26-31 see page 7, line 25-31 see page 17, line 12-28 see page 21, line 36 - page 22, line 28 see page 30, line 25 - page 31, line 8 see page 31, line 26 - page 32, line 12	
	-/--	

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
8 October 1998	19/10/1998
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Dhondt, E

1

Form PCT/ISA/210 (second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
Y	CHANDRANMENON G P ET AL: "TRADING PACKET HEADERS FOR PACKET PROCESSING" IEEE / ACM TRANSACTIONS ON NETWORKING, vol. 4, no. 2, 1 April 1996, pages 141-152, XP000582666 see page 141, right-hand column, line 5-8 see page 141, right-hand column, line 21-23 see page 144, left-hand column, line 7-54 see page 147, left-hand column, line 22-35	6-11, 16-19, 22,23, 25,26
A	TONG-BI PEI ET AL: "VLSI IMPLEMENTATION OF ROUTING TABLES: TRIES AND CAMS" NETWORKING IN THE NINETIES, BAL HARBOUR, APR. 7 - 11, 1991, vol. 2, 7 April 1991, pages 515-524, XP000223375 INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS see page 516, left-hand column, line 12 - right-hand column, line 8	3,12-14, 20,21
A	GB 2 217 488 A (RACAL DATA COMMUNICATIONS INC) 25 October 1989 see figure 18 see page 35, line 5-11 see page 36, line 4-9 see page 38, line 24-25 see page 40, line 16 - page 41, line 9	3,12-14, 20,21
P,X	EP 0 847 217 A (SONY EUROPA B V) 10 June 1998	24
P,A	see abstract see column 17, line 21-28	1-23, 25-30

1

INTERNATIONAL SEARCH REPORT

. .ırmation on patent family members

Interna .l Application No
PCT/SE 98/01289

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
WO 9713377	A	10-04-1997	AU	7089996 A	28-04-1997
			EP	0853864 A	22-07-1998
GB 2217488	A	25-10-1989	US	5202985 A	13-04-1993
EP 0847217	A	10-06-1998	JP	10190699 A	21-07-1998